

# **EXHIBIT E**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

<b>In re U.S. Patent No:</b> 7,714,747	<b>R.C.N.:</b> Not yet assigned
<b>Issued:</b> May 11, 2010	<b>Examiner:</b> Not yet assigned
<b>Applicants:</b> Fallon, James	<b>Art Unit:</b> Not yet assigned
<b>Title:</b> Data Compression Systems and Methods	

**REQUEST FOR *INTER PARTES* REEXAMINATION**

Mail Stop *Inter Partes* Reexam  
Central Reexamination Unit  
Office of Patent Legal Administration  
United States Patent & Trademark Office  
P.O. BOX 1450  
Alexandria, VA 22313-1450

## **TABLE OF CONTENTS**

I.	INTRODUCTION .....	7
A.	Fees .....	7
B.	Certification per 37 C.F.R. § 1.907 that Estoppel Does Not Prohibit the Present <i>Inter Partes</i> Reexamination .....	7
C.	Related Applications .....	8
D.	Related Reexamination Proceedings.....	8
E.	Concurrent Litigation.....	8
F.	Service on the Patent Owner.....	9
G.	Disclaimer Regarding Claim Construction.....	9
II.	CLAIMS REQUESTED .....	9
III.	OVERVIEW OF THE '747 PATENT.....	9
IV.	PROSECUTION HISTORY OF ISSUED CLAIMS .....	12
A.	The Claims of the '747 Patent are only Entitled to a Priority Date of October 29, 2001 .....	12
B.	Prosecution History of the '747 Patent .....	15
C.	Issued Claims of the '747 Patent .....	18
1.	Independent Claim 1 .....	18
2.	Independent Claim 8 .....	19
3.	Independent Claim 14 .....	20
4.	Independent Claim 19 .....	21
V.	IDENTIFICATION OF THE PRIOR ART REFERENCES THAT PRESENT SUBSTANTIAL NEW QUESTIONS OF PATENTABILITY .....	22
VI.	IDENTIFICATION OF THE SNQS AND CORRESPONDING PROPOSED REJECTIONS .....	24
VII.	STATEMENTS POINTING OUT EACH SUBSTANTIAL NEW QUESTION OF PATENTABILITY PURSUANT TO 37 C.F.R. § 1.915(b)(3) .....	26
A.	XMill Discloses a System for Compressing and Decompressing XML-Formatted Data .....	28

1.	Overview of XML-Formatted Data and XMill Compression ..	28
2.	Parsing the Data .....	29
3.	Content-Dependent and Content-Independent Compression ...	30
4.	Identifying the Semantic Compressors Used to Compress the Data and Decompressing .....	32
5.	XMill Discloses Features that Allegedly Distinguished the '747 Patent Over the Prior Art Considered During Initial Examination .....	32
B.	Danskin Discloses a System for Compressing and Decompressing Structured Data .....	34
1.	Overview of the X Graphics Protocol and Structured Data.....	35
2.	Analyzing and Parsing the Data.....	36
3.	Optimized and Default Compressors.....	38
4.	Identifying the Encoder and Decompressing Data .....	39
5.	Danskin Discloses Features that Allegedly Distinguished the '747 Patent Over the Prior Art Considered During Initial Examination .....	40
C.	Sebastian Discloses a Network Architecture Incorporating Format Specific Compression .....	42
1.	Overview of Sebastian .....	42
2.	Analyzing the Data and Providing a Descriptor .....	43
3.	Content-Dependent and Content-Independent Compression ...	43
4.	Sebastian Discloses Features that Allegedly Distinguished the '747 Patent Over the Prior Art Considered During Initial Examination .....	45
D.	Franaszek Discloses an Adaptive Multiple Dictionary Data Compression and Decompression System .....	47
1.	Overview of Franaszek .....	47
2.	Analyzing the Content of the Data and Providing a Descriptor	48
3.	Content-Dependent and Content-Independent Compression ...	50
4.	Franaszek Discloses Features that Allegedly Distinguished the '747 Patent Over the Prior Art Considered During Initial Examination .....	50

E.	RFC 2507 in View of RFC 1144 .....	52
1.	Overview of RFC 2507 Protocol .....	52
2.	Overview of RFC 1144 Protocol .....	54
3.	Motivation to Combine .....	56
4.	RFC 2507 in View of RFC 1144 Discloses Features that Allegedly Distinguished the '747 Patent Over the Prior Art Considered During Initial Examination .....	57
F.	Carr Discloses a Data Processing Compression and Decompression System Including Data Packets .....	58
G.	Dye Discloses Data Compression and Decompression Engines in an Integrated Memory Controller .....	62
H.	Hauck Discloses a Compression Device Using Run Length and Statistical Encoding. ....	65
I.	XML-Xpress Discloses a System for Compressing and Decompressing XML-Formatted Data .....	68
1.	Overview of XML-Xpress .....	68
2.	XML-Xpress Discloses Features that Allegedly Distinguished the '747 Patent Over the Prior Art Considered During Initial Examination .....	70
J.	Seroussi Discloses an Adaptive Data Compression System .....	71
1.	Overview of Seroussi .....	72
2.	Seroussi Discloses Features that Allegedly Distinguished the '747 Patent Over the Prior Art Considered During Initial Examination .....	73
K.	Fascenda Discloses a Realtime Encoding and Transmission Method .....	74
VIII.	DETAILED EXPLANATION OF THE PERTINENCY AND MANNER OF APPLYING THE PATENTS AND PRINTED PUBLICATIONS PURSUANT TO 37 C.F.R. § 1.915(b)(3) .....	76
A.	SNQ 1-5: Proposed Rejections based on XMill as the Primary Reference .....	77
1.	SNQ 1: XMill Anticipates Claims 1-22 under 35 U.S.C. § 102(b) .....	77

2.	SNQ 2: XMill in view of Sebastian Renders Claims 1, 4, 8, 11, 14-15, 19-20 Obvious Under 35 U.S.C. § 103(a).....	106
3.	SNQ 3: XMill and Sebastian and Seroussi Render Claims 5, 6, 12, 16, 17, 21 Obvious Under 35 U.S.C. § 103(a).....	136
4.	SNQ 4: XMill in view of Fascenda Renders Claims 7, 13, 18, and 22 Obvious under 35 U.S.C. § 103(a).....	147
5.	SNQ 5: XMill and XML-Xpress Render Claims 1, 4, 5, 7, 8, and 11-22 Obvious Under 35 U.S.C. § 103(a) .....	154
B.	SNQ 6-7: Proposed Rejections based on Danskin as the Primary Reference .....	202
1.	SNQ 6: Danskin Anticipates Claims 1-22 under 35 U.S.C. § 102(b).....	202
2.	SNQ 7: Danskin in view of XMill Renders Claims 3, 10, 14, and 19 Obvious Under 35 U.S.C. § 103(a).....	242
C.	SNQ 8-10: Proposed Rejections based on Sebastian as the Primary Reference .....	262
1.	SNQ 8: Sebastian Anticipates Claims 1-22 under 35 U.S.C. § 102(e).....	262
2.	SNQ 9: Sebastian in view of Seroussi Renders Claims 5, 12, 16, and 21 Obvious under 35 U.S.C. § 103(a).....	324
3.	SNQ 10: Sebastian in view of Fascenda Renders Claims 7, 13, 18, and 22 Obvious under 35 U.S.C. § 103(a).....	331
D.	SNQ 11-14: Proposed Rejections based on Franaszek as the Primary Reference .....	340
1.	SNQ 11: Franaszek Anticipates Claims 1, 2, 4-9, 11-22 under 35 U.S.C. § 102(b).....	340
2.	SNQ 12: Franaszek in view of Seroussi Renders Claims 5, 12, 16, and 21 Obvious under 35 U.S.C. § 103(a).....	383
3.	SNQ 13: Franaszek in view of Fascenda Renders Claims 7, 13, 18, and 22 Obvious under 35 U.S.C. § 103(a).....	390
4.	SNQ 14: Franaszek in view of XMill Renders Claims 2, 3, 9, and 10 Obvious under 35 U.S.C. § 103(a).....	397
E.	SNQ 15-16: Proposed Rejections based on RFC2507 as the Primary Reference .....	408

1.	SNQ 15: RFC2507 in view of RFC1144 Renders Claims 1, 2, 4-9, 11-22 Obvious under 35 U.S.C. § 103(a) .....	408
2.	SNQ 16: RFC2507 and RFC1144, further in view of XMill, Render Obvious Claims 3 and 10 under 35 U.S.C. § 103(a) ..	462
F.	SNQ 17-18: Proposed Rejections based on Carr as the Primary Reference .....	468
1.	SNQ 17: Carr Anticipates Claims 1-22 under 35 U.S.C. § 102(b) .....	468
2.	SNQ 18: Carr in view of Fascenda Renders Claims 7, 13, 18, and 22 Obvious under 35 U.S.C. § 103(a).....	514
G.	SNQ 19-21: Proposed Rejections based on Dye as the Primary Reference .....	523
1.	SNQ 19: Dye Anticipates Claims 1, 2, 4-9, 11-22 under 35 U.S.C. § 102(e) .....	523
2.	SNQ 20: Dye in view of Seroussi Renders Claims 5, 12, 16, and 21 Obvious under 35 U.S.C. § 103(a).....	569
3.	SNQ 21: Dye in view of XMill Renders Claims 3 and 10 Obvious under 35 U.S.C. § 103(a).....	576
H.	SNQ 22: Proposed Rejections based on Hauck as the Primary Reference .....	581
1.	SNQ 22: Hauck Anticipates Claims 1-22 under 35 U.S.C. § 102(b).....	581
I.	SNQ 23-25: Proposed Rejections based on XML-Xpress as the Primary Reference .....	607
1.	SNQ 23: XML-Xpress Anticipates Claims 1, 4-8, 11-22 under 35 U.S.C. § 102(a) .....	607
2.	SNQ 24: XML-Xpress and Sebastian Render Obvious Claims 1, 2, 4, 8, 9, 11, 14, 15, 19, and 20 Under 35 U.S.C. § 103(a) ...	643
3.	SNQ 25: XML-Xpress and Sebastian, further in view of XMill, Render Obvious Claims 3 and 10 Under 35 U.S.C. § 103(a) .	692
IX.	CONCLUSION.....	696

After loading and unzipping the containers, the decompressor parses the structure container, invokes the corresponding semantic decompressor for the data items and generates the output.

(XMill at p. 9.)

selecting one or more lossless decoders for a data block associated with the data packet, wherein the selecting is based on the descriptor;	<i>(part of claim 1)</i>
---------------------------------------------------------------------------------------------------------------------------------------------	--------------------------

XMill discloses that the decompressor parses the structure container to select the semantic decompressor corresponding to the data items associated with the input data:

After loading and unzipping the containers, the decompressor parses the structure container, invokes the corresponding semantic decompressor for the data items and generates the output.

(XMill at p. 9.)

XMill discloses a lossless configuration embodiment where white space is preserved:

So far we have ignored white spaces between tags, e.g. between <Book> and <Title>, and the decompressor produces a standard indentation: this is sufficient for most applications. Optionally, XMill can preserve the white spaces faithfully: in that case it stores them in container 1.

(XMill at p. 10.)

decompressing the data block with a selected lossless decoder utilizing content dependent data decompression, if the descriptor indicates the data block is encoded utilizing content dependent data compression; and	<i>(part of claim 1)</i>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------

XMill discloses that the compressor performs content-dependent encoding on data:

**Apply different compressors to different containers** Some data items are text, others are numbers, while others may be DNA sequences. XMill applies different specialized compressors (semantic *compressors*) to different containers.

(XMill at p. 2.)

XMill discloses a structure container with information indicating the encoders used by the XMill compressor to compress the data; the XDemill decompressor parses the structure container to identify the corresponding decoders to use for decompression. If a content-



dependent semantic compressor was used to compress the data, a corresponding content-dependent decompressor will be used to decompress:

After loading and unzipping the containers, the decompressor parses the structure container, invokes the corresponding semantic decompressor for the data items and generates the output.

(XMill at p. 9.)

decompressing the data block with a selected lossless decoder utilizing content independent data decompression, if the descriptor indicates the data block is encoded utilizing content independent data compression. <div style="text-align: right;"><i>(part of claim 1)</i></div>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

XMill discloses that the compressor performs content-independent encoding on data by copying it to a container and then using a default gzip compressor:

Users, of course, do not have to specify any semantic compressor: the default text semantic “compressor” simply copies its input to the container, without any semantic compression.

(XMill at p. 9.)

The text compressor *t* does not compress, but rather copies the string to the container unchanged (it will be compressed later by gzip).

(XMill at p. 13.)

XMill discloses a structure container with information indicating the encoders used by the XMill compressor to compress the data. The XDemill decompressor parses the structure container to identify the appropriate decoders to use for decompression. If the structure container indicates that a data item was only copied by the text compressor without compression, then only content-independent unzipping decompression is used for that data item:

After loading and unzipping the containers, the decompressor parses the structure container, invokes the corresponding semantic decompressor for the data items and generates the output.

(XMill at p. 9.)

Thus, as set forth in detail above, XMill teaches each and every feature of independent claim 1. Accordingly, claim 1 is unpatentable under 35 U.S.C. § 102(b) as anticipated by XMill.

performance general-purpose coder, but the huge performance gains of schema-specific encoding are lost.

(XML-Xpress at p. 2.)

XML-Xpress discloses the use of lossless decoders:

Lossless compression as high as 34:1 at throughputs of up to 9 Mbytes/sec were achieved on a test database.

(XML-Xpress at p. 1.)

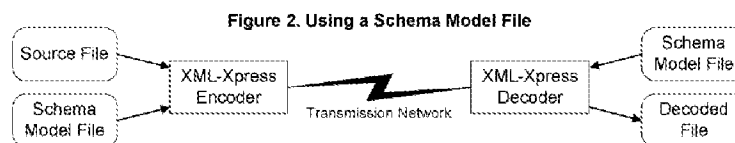
All coders were evaluated in lossless mode - the decoded field had to be a byte-for-byte match with the source files.

(XML-Xpress at p. 5.)

decompressing the data block with a selected lossless decoder utilizing content dependent data decompression, if the descriptor indicates the data block is encoded utilizing content dependent data compression; and

XML-Xpress discloses performing content-dependent data decompression based on the encoders identified by the schemas, if the schema indicates that the file was compressed using content-dependent data compression:

Each SMF supports a specific schema. To compress an XML file, the XML-Xpress real-time encoder needs to have access to the SMF that supports the schema to which the file conforms



The same SMF is provided to both the encoder and the decoder. The real-time encoder can support multiple schemas as long as it has the SMF for each and knows which one to use for each file. This information is available in most XML applications, so that they can take full advantage of the power of schema-specific coding.

(XML-Xpress at p. 3.)

The schema allows the XML tags to be encoded with high efficiency. For example, if an element contains two sub-elements (A,B), the decoder can reconstruct the tags without needing any information from the encoded file: it knows exactly where the start and end tags of both elements A and B will be located.

[K]nowledge of the schema allows the element data to be encoded efficiently. Because schemas supply the data type information, compression routines optimized for specific data types can be used.

(XML-Xpress at p. 2.)

XML-Xpress discloses the use of lossless decompression:

Lossless compression as high as 34:1 at throughputs of up to 9 Mbytes/sec were achieved on a test database.

(XML-Xpress at p. 1.)

All coders were evaluated in lossless mode - the decoded field had to be a byte-for-byte match with the source files.

(XML-Xpress at p. 5.)

decompressing the data block with a selected lossless decoder utilizing content independent data decompression, if the descriptor indicates the data block is encoded utilizing content independent data compression. <div style="text-align: right;"><i>(part of claim 1)</i></div>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

XML-Xpress discloses performing content-independent data decompression using a generic algorithm, if the data block was encoded using the general-purpose, content-independent coder:

If a file does not conform to the expected schema, the data is safely encoded using a high-performance ***general-purpose coder***, but the huge performance gains of schema-specific encoding are lost.

(XML-Xpress at p. 2) (emphasis added).

For environments that support a large number of schemas, XML-Xpress can operate like ICT's flagship product XpressFiles - providing specialized support for the most common schemas and using a generic algorithm for the rest. In this way, the performance benefits of schema-specific coding are obtained within systems serving a diverse range of schemas.

(XML-Xpress at p. 4.)

XML-Xpress discloses the use of lossless decompression:

Lossless compression as high as 34:1 at throughputs of up to 9 Mbytes/sec were achieved on a test database.

(XML-Xpress at p. 1.)

All coders were evaluated in lossless mode - the decoded field had to be a byte-for-byte match with the source files.